Topic Modeling using Topics from Many Domains, Lifelong Learning and Big Data

Zhiyuan Chen CZYUANACM@GMAIL.COM
Department of Computer Science, University of Illinois at Chicago

Bing Liu LIUB@CS.UIC.EDU
Department of Computer Science, University of Illinois at Chicago

Abstract

Topic modeling has been commonly used to discover topics from document collections. However, unsupervised models can generate many incoherent topics. To address this problem, several knowledge-based topic models have been proposed to incorporate prior domain knowledge from the user. This work advances this research much further and shows that without any user input, we can mine the prior knowledge automatically and dynamically from topics already found from a large number of domains. This paper first proposes a novel method to mine such prior knowledge dynamically in the modeling process, and then a new topic model to use the knowledge to guide the model inference. What is also interesting is that this approach offers a novel lifelong learning algorithm for topic discovery, which exploits the big (past) data and knowledge gained from such data for subsequent modeling. Our experimental results using product reviews from 50 domains demonstrate the effectiveness of the proposed approach.

1. Introduction

Topic models, such as LDA (Blei et al., 2003) and pLSA (Hofmann, 1999), have been widely used to discover topics in text documents. Most of such models are unsupervised. However, researchers have shown that unsupervised models may produce many incoherent topics because the objective functions of topic models may not correlate well with human judgments (Chang et al., 2009). To tackle this problem, several *knowledge-based topic models* (KBTM) have been proposed. They use prior domain knowledge specified by the user to guide modeling. DF-LDA (Andrzejewski et al., 2009) is perhaps the earliest KBTM, which can incorporate two forms of prior knowledge from the user: *must-links* and *cannot-links*. A must-link states that two words should belong to the same topic whereas a cannot-link states that two words should not belong to the same topic. We will discuss this and other existing models in section 2.

In this work, we show that much of the prior knowledge from the user can actually be mined automatically (without user input) from a large amount of data in many domains. In most cases, such data is readily available on the Web. This is possible because although every domain is different, there is a fair amount of concept or topic overlapping across domains. For example, every product review domain probably has the topic *price*, reviews of most electronic products share the topic of *battery* and reviews of some products share the topic of *screen*. Topics produced from a single domain can be erroneous (i.e., a topic may contain some irrelevant words in its top ranked positions), but if we can find a set of shared words among some topics generated from multiple domains, these shared words are more likely to be coherent for a particular topic. They can serve as a piece of prior knowledge to help topic modeling in each of these domains or in a new domain.

For example, we have product reviews from three domains. We run LDA to generate a set of topics from each domain. Every domain has a topic about *price*, which is listed below with its top four words (words are ranked based on their probabilities under each topic):

Domain 1: *price*, *color*, *cost*, *life*
Domain 2: *cost*, *picture*, *price*, *expensive*
Domain 3: *price*, *money*, *customer*, *expensive*

These topics are not perfect due to the incoherent words: *color*, *life*, *picture*, and *customer*. However, if we focus on those topical words that appear together in the same topic across at least two domains, we find the following two sets:

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

{*price*, *cost*} and {*price*, *expensive*}.

We can see that the words in such a set are likely to belong to the same topic. Such, {*price*, *cost*} and {*price*, *expensive*}, can serve as *prior knowledge*, which we call *prior knowledge sets* (or *pk-sets* for short), in a KBTM to improve the output topics for each of the three domains or a new domain. For example, after running a KBTM on the reviews of Domain 1, we may find the new topic: *price*, *cost*, *expensive*, *color*, which has three coherent words in the top four positions rather than only two words as in the original topic. This represents a good topic improvement.

The above discussion suggests a three-step approach to our task. Given a set of document corpora $D = \{D_1, \ldots, D_n\}$ from n domains, step 1 runs a topic model (e.g., LDA) on each domain $D_i \in D$ to produce a set of topics S_i. We call these topics the *prior topics* (or *p-topics* for short). Step 2 mines a set of pk-sets (prior knowledge sets) K from all the p-topics $S = \cup_i S_i$. Step 3 uses the pk-sets K in a KBTM to generate topics for a test document collection D^t (D^t may or may not be from D).

To further improve, our proposed method embeds step 2 in step 3 so that the mining of prior knowledge is targeted and thus more accurate. Specifically, we first run a KBTM on the test document collection D^t without any knowledge (which is equivalent to LDA) until its topics (A^t) stabilize. To distinguish these topics from p-topics, we call these topics the *current topics* (or *c-topics* for short). For each c-topic $a_j \in A^t$, we then find a set of matching or similar p-topics M_j^t in S (the set of all p-topics). The intuition here is that these p-topics M_j^t are targeted with respect to a_j and should provide high quality knowledge for a_j. We then mine M_j^t to generate pk-sets K_j^t for c-topic a_j. After that, we continue the execution of the KBTM on D^t, which is now guided by the new pk-sets K^t (which is the union of all K_j^t), in order to generate better c-topics (details are in Section 4).

Regarding knowledge-based topic models (KBTM), we could not use the existing ones because they typically assume the given prior knowledge to be correct (see Section 2). There is clearly no guarantee that the automatically mined pk-sets are all correct for a domain. First, due to wrong topics in S or mining errors, the words in a pk-set may not belong to the same topic in general. Second, the words in a pk-set may belong to the same topic in some domains, but not in others due to the domain diversity. Thus, to apply such knowledge in modeling, the model must deal with possible errors in pk-sets. We propose a new fault-tolerant knowledge-based model to deal with the problem. It can exploit the automatically mined prior knowledge and deal with incorrect knowledge to produce superior topics.

Due to this ability of using topics (or knowledge) generated from other domains to help modeling in the current domain, this work offers two novel capabilities: (1) lifelong learning (Thrun, 1998) and (2) modeling with big data. We call the proposed model *Lifelong Topic Model* (*LTM*).

In summary, this paper makes the following contributions:

1. It proposes a novel approach to exploit text collections from many domains to learn prior knowledge to guide model inference in order to generate more coherent topics. The process is fully automatic. This actually also represents a lifelong learning approach. To our knowledge, it is the first lifelong learning method for topic modeling. It also helps deal with big data, see Section 5.
2. It proposes an effective method to mine/learn quality knowledge dynamically from raw topics produced using text data from a large number of domains.
3. It proposes a new knowledge-based topic model LTM that is able to deal with possible incorrect knowledge.

2. Related Work

The most related work to this paper is the knowledge-based topic models (KBTM), (e.g., Andrzejewski et al. 2009; Chen et al. 2013b; Mukherjee & Liu 2012). However, the models in these papers typically assume that the given knowledge is correct. The GK-LDA model in (Chen et al., 2013a) has an automated mechanism to deal with wrong lexical knowledge by using the ratio of probabilities of two words under each topic to estimate the reliability of each piece of knowledge. In contrast, the proposed LTM model dynamically balances the use of learned knowledge and the information in the actual document collection during the Gibbs sampling. As we will see in Section 5, LTM outperforms both DF-LDA and GK-LDA significantly. LTM also automatically mines the prior knowledge. None of the above existing models can do that. In (Chen et al., 2014), we proposed a preliminary method for the same task. The LTM model has the additional mechanism to dynamically adjust (and re-extract) knowledge that targets the current model. The fault tolerant ability of LTM is also shown to be more effective in Section 5.

Besides the above knowledge-based topic models, Blei & McAuliffe (2007) and Ramage et al. (2009) studied topic models in a supervised setting. Ahmed et al. (2013) considered location knowledge in topic models. All of these models assume that the prior knowledge, e.g., labels or regions, is correct. Their modeling tasks are also quite different from ours.

Our work is also related to transfer learning and lifelong learning. Topic models have been used to help transfer learning (Xue et al., 2008; Pan & Yang, 2010). However, transfer learning in these papers is for traditional supervised classification. Kang et al. (2012) proposed topic models with better fitting by transferring labels. We do not use

any labeled data. Yang et al. (2011) modeled the language gap between topics using a user provided parameter indicating the degree of technicality of the domain. In contrast, our proposed LTM model is fully automatic. Our work also touches multi-domain learning (Joshi et al., 2012). However, multi-domain learning is still for supervised classification. Our focus is topic modeling. In the context of topic modeling, we are not aware of any existing model for lifelong learning (Thrun, 1998; Silver et al., 2013) .

Last, since we use product reviews as our data, our work is related to opinion mining using topic models, e.g., (Mei et al., 2007; Titov & McDonald, 2008; Zhao et al., 2010). However, none of the models mines prior knowledge and uses knowledge to guide modeling.

3. Overall Algorithm

This section first introduces the proposed overall algorithm, which follows the idea described in Section 1. It then introduces a lifelong learning approach for topic modeling. The algorithm consists of two general steps:

Step 1 (**prior topic generation**): Given a set of document collections $D = \{D_1, \ldots, D_n\}$ from n domains, Algorithm 1 **PriorTopicsGeneration** runs LDA on each domain $D_i \in D$ to produce a set of topics S_i (lines 2 and 4). The resulting topics from all n domains are unionized together to produce the set of all topics S (line 9) from D. We call S the *prior topic* (or *p-topic*) set. The *p-topics* in S are used in the proposed model LTM to generate the prior knowledge.

Iterative improvement: The above process can actually be run iteratively to improve the p-topics in S. That is, S from the previous iteration can help generate better topics from D using the proposed LTM model for the next iteration. This process is reflected in lines 1, 5-7 and 10. We will examine the performance of different iterations in Section 5.2. Note that from the second iteration ($r \geq 1$), LTM is used (line 6).

Step 2 (**testing**): Given a test document collection D^t and a prior topic (or p-topic) set S, this step employs the proposed topic model LTM (Algorithm 2) to generate topics from D^t. To distinguish these topics from p-topics, we call them the *current topics* (or *c-topics* for short). LTM is given in Algorithm 2, which we will detail in the next section. Note that D^t can be a document collection from D or a new domain. This can be seen as two ways of using the proposed algorithm: (1) the topics from D^t can be part of p-topics in S used in knowledge mining in LTM, and (2) not part of p-topics in S. We will experiment with these two settings in Section 5.

Lifelong learning: The above approach naturally enables lifelong learning. S is the knowledge base (e.g., the p-topic set) generated by a system (or even specified by the user), and LTM is the learning algorithm. Given a new learning task G (e.g., topic modeling in our case) with its data (e.g., D^t), lifelong learning works in two main phases.

Algorithm 1 PriorTopicsGeneration(D)

```
1:  for r = 0 to R do
2:    for each domain corpus D_i ∈ D do
3:      if r = 0 then
4:        S_i ← LDA(D_i);
5:      else
6:        S_i ← LTM(D_i, S);
7:      end if
8:    end for
9:    S ← ∪_i S_i;
10: end for
```

Algorithm 2 LTM(D^t, S)

```
1: A^t ← GibbsSampling(D^t, ∅, N); // Run N Gibbs iterations with no knowledge (equivalent to LDA).
2: for i = 1 to N do
3:   K^t ← KnowledgeMining(A^t, S);
4:   A^t ← GibbsSampling(D^t, K^t, 1); // Run with knowledge K^t.
5: end for
```

Phase 1: **Learning with prior knowledge**: This is essentially Step 2 above using LTM, which solves two subproblems. Step 1 is the initialization.

a) *Identify shared knowledge for task G*. Identify the part of the knowledge in S that can be used for G. In our case, the shared knowledge is K^t in Algorithm 2.
b) *Knowledge-based learning*. Learn for task G with the help of K^t using a learning algorithm. In our case, it is the GibbsSampling function in line 4 of LTM.

Phase 2: **Knowledge retention and consolidation**. In our case, we simply add the topics from G to S if G is a new task. If G is an old task, we replace its topics in S. This is not included in Algorithms 1 or 2, but can be added easily.

4. LTM Model

Like many topic models, LTM uses Gibbs sampling for inference (Griffiths & Steyvers, 2004). Its graphical model is the same as LDA, but LTM has a very different sampler which can incorporate prior knowledge and also handle errors in the knowledge.

LTM works as follows: It first runs the Gibbs sampler of LTM for N iterations (or sweeps) to find a set of initial topics A^t from D^t with no knowledge (line 1, Algorithm 2). Since there is no knowledge, the sampler is equivalent to that of LDA. It then makes another N Gibbs sampling sweeps (lines 2-5). But in each of these new sweeps, it first mines pk-sets K^t for all topics in A^t using the function

Algorithm 3 KnowledgeMining(A^t, S)

1: **for** each p-topic $s_k \in S$ **do**
2: $j^* = \min_j$ KL-Divergence(a_j, s_k) for $a_j \in A^t$;
3: **if** KL-Divergence(a_{j^*}, s_k) $\leq \pi$ **then**
4: $M^t_{j^*} \leftarrow M^t_{j^*} \cup \{s_k\}$;
5: **end if**
6: **end for**
7: $K^t \leftarrow \cup_{j^*}$ FIM($M^t_{j^*}$); // Frequent Itemset Mining.

KnowledgeMining (Algorithm 3, detailed in Section 4.1) and then uses K^t to generate a new set of topics from D^t. The intuition of LTM has been explained in Section 1. Note that to make the algorithm more efficient, we do not need to mine knowledge for every sweep (see Section 5.5). Below, we focus on the knowledge mining function of LTM. The Gibbs sampler will be given in Section 4.2.

4.1. Knowledge Mining

The knowledge-mining function is given in Algorithm 3. For each p-topic $s_k \in S$, it finds the best matching (or the most similar) c-topic a_{j^*} in the c-topic set A^t (line 2). $M^t_{j^*}$ is used to mine pk-sets for c-topic a_{j^*} (line 7). We find the matching p-topics for each individual c-topic a_{j^*} because we want a_{j^*} specific p-topics for more accurate knowledge set mining. Below, we present the algorithms for topic match and knowledge set mining.

Topic match (lines 2-5, Algorithm 3): To find the best match for s_k with a c-topic a_{j^*} in A^t, we use KL Divergence to compute the difference of the two distributions (lines 2 and 3). In this paper, we use Symmetrised KL Divergence for all divergence computing, i.e., given two distributions P and Q, the divergence is calculated as $(KL(P,Q) + KL(Q,P))/2$. We denote the c-topic with the minimum KL Divergence with s_k as a_{j^*}. π is used to ensure the p-topics in $M^t_{j^*}$ are sufficiently correlated with a_{j^*}.

Mine knowledge sets using frequent itemset mining (FIM): Given p-topics in each matched set $M^t_{j^*}$, this step finds sets of words that appear together multiple times in these p-topics. Each set of shared words among related p-topics across multiple domains are likely to belong to the same topic. To find such sets of words in the matched set of p-topics $M^t_{j^*}$, we use frequent itemset mining (FIM) (Agrawal & Srikant, 1994).

FIM is stated as follows: Given a set of transactions X, where each transaction $x_i \in X$ is a set of items. In our context, x_i is a set of top words of a p-topic (no probability attached). X is actually $M^t_{j^*}$ without lowly ranked words in each p-topic as only the top words are usually representative of a topic. The goal of FIM is to find every itemset (a set of items) that satisfies some user-specified frequency threshold (also called *minimum support*), which is the minimum number of times that an itemset should appear in X. Such itemsets are called *frequent itemsets*. In our context, a frequent itemset is a set of words that have appeared together multiple times in the p-topics of $M^t_{j^*}$. Such itemsets are our prior knowledge pk-sets.

In this work, we use only frequent itemsets of length two, i.e., each pk-set has only two words. For example, {battery, life}, {battery, power}, {battery, charge}. Using two words in a pk-set is sufficient to cover the semantic relationship of words belonging to the same topic. Longer sets tend to contain more errors since some words in a set may not belong to the same topic as others. Such errors can hurt the downstream modeling.

4.2. Gibbs Sampler

This sub-section gives the Gibbs sampler of the LTM model, which differs from LDA as LTM needs additional mechanisms to leverage the prior knowledge and to also deal with wrong knowledge during sampling. Below, we first discuss the techniques used for these two capabilities, and then present the final Gibbs sampler.

4.2.1. Incorporating Prior Knowledge and Dealing with Wrong Knowledge

As each pk-set reflects a possible semantic similarity relation between a pair of words, we use the *generalized Pólya urn* (GPU) model (Mahmoud, 2008) to leverage this knowledge in Gibbs sampling to encourage the pair of words to appear in the same topic. GPU was first introduced to topic modeling in (Mimno et al., 2011). However, their model is not concerned with any prior knowledge, and is still an unsupervised model.

The *Pólya urn* model works on colored balls and urns. In the topic model context, a word can be seen as a ball of a certain color and a topic as an urn. The classic topic-word distribution is reflected by the color proportion of balls in the urn. LDA follows the *simple Pólya urn* (SPU) model in the sense that when a ball of a particular color is drawn from an urn, that ball is put back to the urn along with a new ball of the same color. The content of the urn changes over time, which gives a self-reinforcing property known as "the rich get richer". The GPU model differs from SPU in that, when a ball of a certain color is drawn, two balls of that color is put back along with a certain number of balls of some other colors. These additional balls of some other colors added to the urn increase their proportions in the urn. We call this the promotion of these colored balls. Applying the idea to our case, when a word w is assigned to a topic t, each word w' that shares a pk-set of topic t with w is also assigned the topic t by a certain amount, which is decided by the matrix $A'_{t,w',w}$. w' is thus promoted by w, meaning that the probability of w' under topic t is also increased.

Here, a pk-set of a topic t means this pk-set is extracted from the p-topics matching with topic t.

The problem is how to set proper values for matrix $A_{t,w',w}$. To answer this question, let us also consider the problem of wrong knowledge. Since the pk-sets are mined from p-topics in multiple previous domains automatically, the semantic relationship of words in a pk-set may not be correct for the current domain. It is a challenge to determine which pk-set is not appropriate. One way to deal with both problems is to assess how the words in a pk-set correlated with each other in the current domain. If they are more correlated, they are more likely to be correct for a topic in the domain and thus should be promoted more. If they are less correlated, they are more likely to be wrong and should be promoted less (or even not promoted).

To measure the correlation of two words in a pk-set in the current domain, we use Pointwise Mutual Information (PMI), which is a popular measure of words association in text. It has also been used to evaluate topic models (Newman et al., 2010). PMI is the logarithmic ratio of the actual joint probability of two events to the expected joint probability if the two events were independent. In our case, it measures the extent to which two words tend to co-occur, which corresponds to the higher-order co-occurrence on which topic models are based (Heinrich, 2009). The PMI of two words is defined as follows:

$$PMI(w_1, w_2) = log\frac{P(w_1, w_2)}{P(w_1)P(w_2)} \quad (1)$$

where $P(w)$ denotes the probability of seeing word w in a random document, and $P(w_1, w_2)$ denotes the probability of seeing both words co-occurring in a random document. These probabilities are empirically estimated from the current domain collection D^t:

$$P(w) = \frac{\#D^t(w)}{\#D^t} \quad (2)$$

$$P(w_1, w_2) = \frac{\#D^t(w_1, w_2)}{\#D^t} \quad (3)$$

where $\#D^t(w)$ is the number of documents in D^t that contain the word w and $\#D^t(w_1, w_2)$ is the number of documents that contain both words w_1 and w_2. $\#D^t$ is the total number of documents in D^t. A positive PMI value implies a true semantic correlation of words, while a non-positive PMI value indicates little or no semantic correlation. Thus, we only consider pk-sets with positive PMI values. We also add a parameter factor μ to control how much the GPU model should trust the word relationships indicated by PMI (see the setting of μ in Section 5.1). Finally, the amount of promotion for word w' when seen w is defined as follows:

$$A'_{t,w',w} = \begin{cases} 1 & w = w' \\ \mu \times PMI(w, w') & (w, w') \text{ is a pk-set of } t \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

4.2.2. Conditional Distribution of Gibbs Sampler

The GPU model is nonexchangeable, i.e., the joint probability of the words in any given topic is not invariant to the permutation of those words. The inference for the model can be computationally expensive due to the non-exchangeability of words, that is, the sampling distribution for the word of interest depends on each possible value for the subsequent words along with their topic assignments. We take the approach of (Mimno et al., 2011) which approximates the true Gibbs sampling distribution by treating each word as if it were the last. The approximate Gibbs sampler has the following conditional distribution:

$$P(z_i = t|\boldsymbol{z}^{-i}, \boldsymbol{w}, \alpha, \beta, A') \propto \frac{n^{-i}_{d,t} + \alpha}{\sum_{t'=1}^{T}(n^{-i}_{d,t'} + \alpha)} \times \frac{\sum_{w'=1}^{V} A'_{t,w',w_i} \times n^{-i}_{t,w'} + \beta}{\sum_{v=1}^{V}(\sum_{w'=1}^{V} A'_{t,w',v} \times n^{-i}_{t,w'} + \beta)} \quad (5)$$

where n^{-i} is the count excluding the current assignment of z_i, i.e., $\boldsymbol{z}^{-i}$, $\boldsymbol{w}$ refers to all the words in all documents in the document collection and w_i is the current word to be sampled with a topic denoted by z_i. $n_{d,t}$ denotes the number of times that topic t was assigned to words in document d, which is the document index of word w_i. $n_{t,v}$ refers to the number of times that word v appears under topic t. α and β are predefined Dirichlet hyper-parameters. T is the number of topics, and V is the vocabulary size. A' is the promotion matrix defined in Equation 4.

5. Evaluation

This section evaluates the proposed LTM model and compares it with four state-of-the-art baselines:

LDA (Blei et al., 2003): An unsupervised topic model.
DF-LDA (Andrzejewski et al., 2009): A knowledge-based topic model that can use the user-provided knowledge.
GK-LDA (Chen et al., 2013a): A knowledge-based topic model that uses the ratio of word probabilities under each topic to reduce the effect of wrong knowledge.
AKL (Chen et al., 2014): A knowledge-based topic model that applies clustering to learn the knowledge and utilizes the knowledge in the form of knowledge clusters.

Note that although both DF-LDA and GK-LDA can take prior knowledge from the user, they cannot mine any prior knowledge, which make them not directly comparable with LTM. Thus, we have to feed them the knowledge produced using the proposed knowledge mining algorithm. This allows us to assess the knowledge handling capability of each model. AKL uses its own way to generate and incorporate knowledge.

5.1. Experimental Settings

Dataset. We have created a large dataset containing 50 review collections from 50 product domains crawled from

Amazon.com. Each domain has 1,000 (1K) reviews. We followed (Chen et al., 2013a) to pre-process the dataset. The dataset and the code are publically available at the authors' websites. To test the behaviors of LTM for large datasets (see Sections 5.4 and 5.5), we created 4 large review collections with 10,000 (10K) reviews in each. Note that most product domains in our collections do not have such a large number of reviews.

Parameter Setting. For all models, posterior estimates of latent variables were taken with a sampling lag of 20 iterations in the post burn-in phase (first 200 iterations for burn-in) with 2,000 iterations in total. The parameters of all topic models are set as $\alpha = 1$, $\beta = 0.1$, $T = 15$. The other parameters for baselines were set as suggested in their original papers. For parameters of LTM, the top 15 words of each topic were used to represent the topic in the topic matching process and also frequent itemset mining. This is intuitive as the top words in each topic are more likely to be semantically coherent while words at lower positions are much less related. The minimum support threshold is empirically set to $max(5, 0.4 \times \#Trans)$ where $\#Trans$ is the size of each M_{j*}^{t} (Section 4.1). This is also intuitive as appearances in a reasonable number of domains show likely word semantic correlations. The parameter π in Algorithm 3 is empirically set to 7.0. The parameter μ in Equation 4 is set to 0.3, which determines the extent of promotion of words in a pk-set using the GPU model. Intuitively, a too small value of μ will lead to an inferior performance as it basically ignores the knowledge, while a too large value can damage the model too due to the errors in the knowledge.

Test Settings: We use two test settings to evaluate LTM, which represent two ways of using LTM in Section 3:

1. Mine prior knowledge pk-sets from topics of all domains including the test domain.
2. Mine prior knowledge pk-sets from topics of all domains excluding the test domain.

Setting 1 has a slight advantage as in mining knowledge for a test domain collection, its own initial topics are used, which can help find more targeted knowledge. We report the results for Setting 1 in Sections 5.2 and 5.3, and the results for Setting 2 in Section 5.4.

5.2. Topic Coherence of Test Setting 1

This sub-section evaluates the topics generated by each model based on the Topic Coherence measure in (Mimno et al., 2011). Traditionally, topic models are typically evaluated using perplexity. However, as shown in (Chang et al., 2009), the perplexity measure does not reflect the semantic coherence of individual topics, which can sometimes be contrary to the human judgments. The Topic Coherence

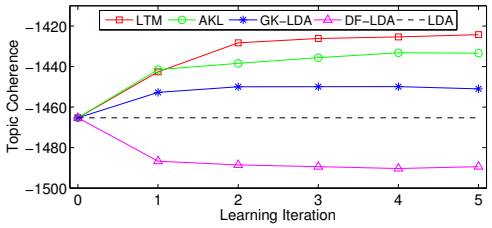

Figure 1. Average Topic Coherence values of each model at different learning iterations for Setting 1 (Iteration 0 = LDA).

measure (Mimno et al., 2011) is proposed as a better alternative for assessing topic quality. It is shown in (Mimno et al., 2011) that Topic Coherence correlates well with human expert labeling. Recently, it has become popular to use Topic Coherence to evaluate topic models (Arora et al., 2013). Furthermore, as our goal is to discover meaningful or coherent topics, Topic Coherence is more suitable for our evaluation. A higher Topic Coherence value indicates a higher quality of topics.

Our proposed algorithm (Algorithm 1) is designed for iterative improvements, i.e., a higher quality of topics can generate better knowledge, which in turn helps discover more coherent topics. This framework is also suitable for DF-LDA, GK-LDA, and AKL, i.e., the topics learned from a model at iteration r is used to generate knowledge for that model at iteration $r + 1$. Iteration 0 is equivalent to LDA (without any knowledge). We call each of these iterations a learning iteration. Since DF-LDA and GK-LDA cannot mine any prior knowledge, they use our proposed knowledge mining method. Our knowledge in the form of pairs (sets of two words) has the same meaning as the knowledge used in DF-LDA (must-link) and GK-LDA (LR-set). In this work, we do not use cannot-links.

Figure 1 shows the average Topic Coherence value of each model at each learning iteration. Each value is the average over all 50 domains. Note that since LDA cannot use any prior knowledge, its results remain the same. From Figure 1, we can see that LTM performs the best and has the highest Topic Coherence values in general. These show that LTM finds higher quality topics than the baselines. Both AKL and GK-LDA perform better than LDA but worse than LTM, showing their ability of dealing with wrong knowledge to some extent. DF-LDA does not perform well. Without an automated way to deal with each piece of (correct or incorrect) knowledge specifically for each individual domain, its performance is actually worse than LDA.

In summary, we can say that the proposed LTM model can generate better quality topics than all baseline models. Even though DF-LDA and GK-LDA use our method for knowledge mining, without an effective wrong knowledge handling method, they are not sufficient. The improvements of LTM are all significant ($p < 0.01$ over AKL and

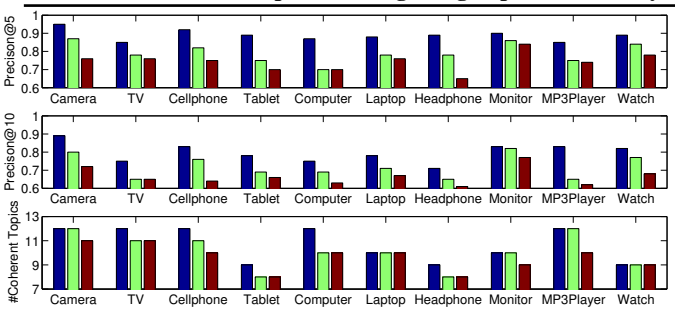

Figure 2. Top & Middle: Topical words $Precision@5$ & $Presicion@10$ of coherent topics of each model respectively; Bottom: number of coherent (#Coherent) topics discovered by each model. The bars from left to right in each group are for LTM, LDA, and DF-LDA. On average, for $Precision@5$ and $Precision@10$, LTM improves LDA by 10% and 8%, and DF-LDA by 15% and 14% respectively. On average, LTM also discovers 0.6 more coherent topics than LDA and 1.1 more coherent topics than DF-LDA over the 10 domains.

$p < 0.0001$ over the other baselines) based on paired t-test.

5.3. Human Evaluation

Here we want to evaluate the topics based on human judgment. The results are still from test Setting 1. Two human judges who are familiar with Amazon products and reviews were asked to label the generated topics. Since we have a large number of domains, we selected 10 domains for labeling. The selection was based on the knowledge of the products of the two human judges. Without enough knowledge, the labeling will not be reliable. We labeled the topics generated by LTM, LDA and DF-LDA at learning iteration 1. For labeling, we followed the instructions in (Mimno et al., 2011).

Topic Labeling. We first asked the judges to label each topic as *coherent* or *incoherent*. The models that generated the topics for labeling were obscure to the judges. In general, a topic was labeled as *coherent* if its topical words are semantically coherent and together represent a semantic concept; otherwise *incoherent*.

Word Labeling. The topics that are labeled as coherent by both judges were used for word labeling. Each topical word was labeled as *correct* if it is coherently related to the concept represented by the topic (identified in the topic labeling step); otherwise *incorrect*.

The Cohens Kappa agreement scores for topic labeling and word labeling are 0.862 and 0.857 respectively.

Evaluation measures. Since topics are rankings of words based on their probabilities, without knowing the exact number of correct topical words, a natural way to evaluate these rankings is to use $Precision@n$ (or $p@n$) which was also used by other researchers, e.g., (Zhao et al., 2010), where n is a rank position. Apart from $p@n$, we also report the number of coherent topics found by each model.

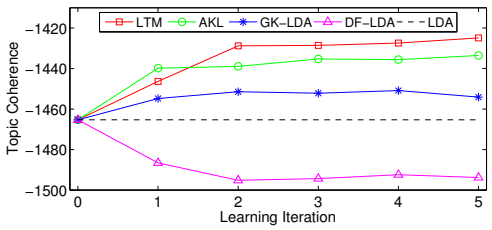

Figure 3. Average Topic Coherence values of each model at different learning iterations in Setting 2. The results are slightly worse than those of Setting 1 (Figure 1).

Figure 2 gives the average topical words $Precision@5$ (top chart) and $Precision@10$ (middle chart) of only coherent topics (those incoherent topics are not considered) for each model in each domain. It is clear that LTM achieves the highest $p@5$ and $p@10$ values in all 10 domains. LDA is slightly better than DF-LDA in general, but clearly inferior to LTM. This is consistent with the Topic Coherence results in Section 5.2. The improvements of LTM vary in domains. For some domains, e.g., Camera, Tablet and Headphone, LTM achieves marked improvements. We found that these domains tend to have a lot of topic overlapping with many other domains. On the other hand, the improvements in the Monitor domain are less because of less topic overlapping with other domains. Significance testing using paired t-test shows that the improvements of LTM over the baselines on $p@5$ and $p@10$ are both significant ($p < 0.0001$). The bottom chart of Figure 2 shows that LTM also discovers more coherent topics than LDA and DF-LDA.

We can then conclude that LTM is superior to the baselines based on both Topic Coherence and human judgment.

5.4. Topic Coherence of Test Setting 2

We now evaluate LTM in Test Setting 2. That is, in mining pk-sets, we do not use the topics from the current domain but only p-topics from the other domains. We set the minimum support threshold for knowledge mining to one less than that for Setting 1 as the current topics are not used. Here we also experiment the iterative process. We use each of the 50 domains as the current domain and the rest 49 domains as the prior domains. Figure 3 shows the average Topic Coherence values for this set of experiments. We see that LTM again achieves higher Topic Coherence values in general, which is consistent with the results in previous sections. The results of LTM (and other knowledge-based models) are slightly worse than those of Setting 1 (Figure 1). This is expected as it does not use its own topics in knowledge mining, which can help mine more suitable knowledge for the domain.

Applying knowledge to 10K reviews. Figures 1 and 3 showed that LTM improves topics for 1,000 (1K) reviews.

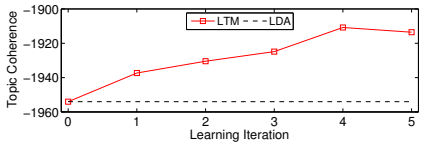

Figure 4. Average Topic Coherence values at different learning iterations over four 10K domains. The knowledge is mined from 49 domains of 1K reviews.

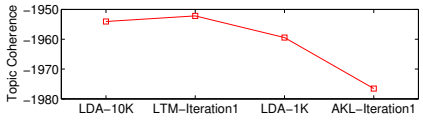

Figure 5. Average Topic Coherence values of each model when dividing big data into small data.

An interesting question is whether LTM can also improve on 10K reviews given that LDA should perform better with 10,000 (10K) reviews as more data give more reliable statistics. We then apply the knowledge learned from test setting 2 at each learning iteration on each of four domains with 10K reviews. Figure 4 gives the average Topic Coherence values over these four domains. We can see that with larger datasets, LTM still gets significant improvements over LDA ($p < 0.0001$ based on paired t-test).

5.5. Improving topic modeling for Big Data

This sub-section shows that our approach can also be exploited to make topic modeling on a single big data more effective, slight improvements in topic quality and major improvements in efficiency.

Following our approach of learning from multiple domains, we divide a big dataset into a number of small datasets and pretend that they are from multiple domains. With multiple small datasets, we can run our experiments just like that in Section 5.2. Here we use each of the four large data sets (10K reviews). Although our four large datasets are not particularly large, as it is shown in (Arora et al., 2013) that LDA using Gibbs sampling is linear in the number of documents, our results here are sufficient to show the trend.

For these experiments, we divide each of our four 10K review collections into 10 folders where each folder has 1K reviews. Then, we run the LTM model treating 10 folders as 10 domains, and evaluate both topic quality and efficiency based on Test Setting 1. Here, we also include AKL in the comparison as it gives the best Topic Coherence among baselines. Note that both PMI in LTM and co-document frequency ratio used in AKL are computed using 10K reviews. Figure 5 shows the Topic Coherence value of each model. Topic Coherence is calculated using 10K reviews. We can see that LTM achieves slightly higher Topic Coherence than LDA-10K (LDA on 10K reviews) and much higher Topic Coherence than LDA-1K (LDA on 1K reviews). AKL, however, gets the lowest Topic Coherence. We investigated its results and found that for the

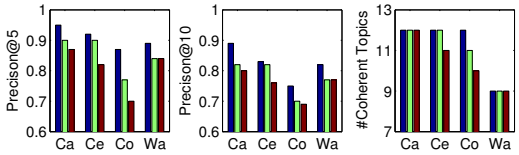

Figure 6. Human labeling of LTM, LDA-10K and LDA-1K (bars from left to right). The domains from left to rights are Camera, Cellphone, Computer, and Watch.

noisy topics AKL tends to group them into the clusters of good topics, which lowers the quality of the mined knowledge. For AKL, we also tried different numbers of clusters with no improvements. The knowledge mining method in LTM is shown to be more effective. Since the 10 folders contain similar information, one learning iteration is sufficient (more learning iterations gave quite similar results). We also employ human labeling as in Section 5.3. For LTM and LDA-1K, we labeled the folder with the highest Topic Coherence value. The results are given in Figure 6 which also shows a slightly better performance of LTM. The improvement of the labeled folder of LTM is 17 points compared with LDA-10K in terms of Topic Coherence. The topic quality improvements are not large due to the fact that the 10 small datasets are from the same domain and are thus less effective for knowledge learning.

LTM's running time is about 31% of LDA-10K because we can run the 10 folders in parallel. Note that LTM in Figure 5 updates knowledge in every 50 iterations. LTM can be easily applied in MapReduce to further solve the memory issue with the big data.

In summary, we can conclude that with our LTM model, it is possible to run a big data set by dividing it into smaller datasets and achieve slightly better topic quality while greatly reduce the execution time.

6. Conclusions

This paper studied the problem of using data from a large number of domains to mine useful prior knowledge to guide topic modeling in order to generate more coherent topics. It proposed an advanced topic model LTM that not only mines prior knowledge automatically, but also exploits the mined prior knowledge to generate better topic results. Additionally, the paper showed that LTM actually represents a novel lifelong learning algorithm for topic discovery. It can be further exploited to deal with topic modeling in big data. Experimental results using product review collections from 50 domains demonstrated the effectiveness of the proposed LTM model.

Acknowledgments

This work was supported in part by a grant from National Science Foundation (NSF) under grant no. IIS-1111092.

References

Agrawal, Rakesh and Srikant, Ramakrishnan. Fast algorithms for mining association rules. In *VLDB*, volume 1215, pp. 487–499, 1994.

Ahmed, Amr, Hong, Liangjie, and Smola, Alexander J. Nested Chinese Restaurant Franchise Process: Applications to User Tracking and Document Modeling. In *ICML*, pp. 1426–1434, 2013.

Andrzejewski, David, Zhu, Xiaojin, and Craven, Mark. Incorporating domain knowledge into topic modeling via Dirichlet Forest priors. In *ICML*, pp. 25–32, 2009.

Arora, Sanjeev, Ge, Rong, Halpern, Yonatan, Mimno, David, Moitra, Ankur, Sontag, David, Wu, Yichen, and Zhu, Michael. A Practical Algorithm for Topic Modeling with Provable Guarantees. In *ICML*, pp. 280–288, 2013.

Blei, David M. and McAuliffe, Jon D. Supervised Topic Models. In *NIPS*, pp. 121–128, 2007.

Blei, David M., Ng, Andrew Y., and Jordan, Michael I. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Chang, Jonathan, Boyd-Graber, Jordan, Chong, Wang, Gerrish, Sean, and Blei, David M. Reading Tea Leaves: How Humans Interpret Topic Models. In *NIPS*, pp. 288–296, 2009.

Chen, Zhiyuan, Mukherjee, Arjun, Liu, Bing, Hsu, Meichun, Castellanos, Malu, and Ghosh, Riddhiman. Discovering Coherent Topics Using General Knowledge. In *CIKM*, pp. 209–218, 2013a.

Chen, Zhiyuan, Mukherjee, Arjun, Liu, Bing, Hsu, Meichun, Castellanos, Malu, and Ghosh, Riddhiman. Exploiting Domain Knowledge in Aspect Extraction. In *EMNLP*, pp. 1655–1667, 2013b.

Chen, Zhiyuan, Mukherjee, Arjun, and Liu, Bing. Aspect Extraction with Automated Prior Knowledge Learning. In *ACL*, 2014.

Griffiths, Thomas L and Steyvers, Mark. Finding Scientific Topics. *PNAS*, 101 Suppl:5228–5235, 2004.

Heinrich, Gregor. A Generic Approach to Topic Models. In *ECML PKDD*, pp. 517 – 532, 2009.

Hofmann, Thomas. Probabilistic Latent Semantic Analysis. In *UAI*, pp. 289–296, 1999.

Joshi, Mahesh, Cohen, William W, Dredze, Mark, and Rosé, Carolyn P. Multi-domain Learning: When Do Domains Matter? In *EMNLP*, pp. 1302–1312, 2012.

Kang, Jeon-hyung, Ma, Jun, and Liu, Yan. Transfer Topic Modeling with Ease and Scalability. In *SDM*, pp. 564–575, 2012.

Mahmoud, Hosam. *Polya Urn Models*. Chapman & Hall/CRC Texts in Statistical Science, 2008.

Mei, Qiaozhu, Ling, Xu, Wondra, Matthew, Su, Hang, and Zhai, ChengXiang. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW*, pp. 171–180, 2007.

Mimno, David, Wallach, Hanna M., Talley, Edmund, Leenders, Miriam, and McCallum, Andrew. Optimizing semantic coherence in topic models. In *EMNLP*, pp. 262–272, 2011.

Mukherjee, Arjun and Liu, Bing. Aspect Extraction through Semi-Supervised Modeling. In *ACL*, pp. 339–348, 2012.

Newman, David, Lau, Jey Han, Grieser, Karl, and Baldwin, Timothy. Automatic evaluation of topic coherence. In *HLT-NAACL*, pp. 100–108, 2010.

Pan, Sinno Jialin and Yang, Qiang. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

Ramage, Daniel, Hall, David, Nallapati, Ramesh, and Manning, Christopher D. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, pp. 248–256, 2009.

Silver, Daniel L, Yang, Qiang, and Li, Lianghao. Lifelong Machine Learning Systems: Beyond Learning Algorithms. In *AAAI Spring Symposium: Lifelong Machine Learning*, 2013.

Thrun, Sebastian. Lifelong Learning Algorithms. In Thrun, S and Pratt, L (eds.), *Learning To Learn*. Kluwer Academic Publishers, 1998.

Titov, Ivan and McDonald, Ryan. Modeling online reviews with multi-grain topic models. In *WWW*, pp. 111–120, 2008.

Xue, GR, Dai, Wenyuan, Yang, Q, and Yu, Y. Topic-bridged PLSA for cross-domain text classification. In *SIGIR*, pp. 627–634, 2008.

Yang, Shuang Hong, Crain, Steven P, and Zha, Hongyuan. Bridging the Language Gap: Topic Adaptation for Documents with Different Technicality. In *AISTATS*, volume 15, pp. 823–831, 2011.

Zhao, Wayne Xin, Jiang, Jing, Yan, Hongfei, and Li, Xiaoming. Jointly Modeling Aspects and Opinions with a MaxEnt-LDA Hybrid. In *EMNLP*, pp. 56–65, 2010.